What Is Claimed Is:

1. A method of safeguarding at least one program part that is critical to safety against inadvertent execution, comprising:

executing the at least one program part in a predetermined chronological sequence;

at a certain point in time in the execution, generating a pattern; and

at least at one point in time, checking whether the pattern is present.

2. The method of claim 1, wherein the pattern is generated at a beginning of the execution of the at least one program part.

3. The method of claim 1, wherein the pattern is generated in a volatile memory element.

4. The method of claim 1, further comprising:

checking an external boundary condition at the time of pattern generation and pattern checking.

5. The method of claim 4, wherein a state of a hardware component serves as the external boundary condition.

6. A method of safeguarding a program part that is critical to safety, comprising:

performing a check at least at one point in time during an execution of the program part that is critical to safety to determine a presence of a pattern representing a proper sequence of the program part; and

terminating the execution of the program part if the pattern is determined to be not present.

7. A memory device for storing program instructions to cause a microprocessor to safeguard at least one program part that is critical to safety against inadvertent execution, the microprocessor being divided into at least one area, each area storing a respective one of the at least one program part, the at least one program part being executable in a predetermined chronological sequence, the memory device comprising:

a first arrangement for generating a pattern at a certain point in time when the at least one program part is executed; and

at least one second arrangement for performing a check at a later point in time to determine whether the pattern is present.

8. The memory device of claim 7, further comprising:

an arrangement for resetting the microprocessor.